

**Amendments to the Specification:**

Please replace paragraph [0050] with the following rewritten paragraph:

~~Operation~~ Operational log 440 may include a persistent historical record of critical metadata changes, such as changes to namespace data 410 and mapping data 420. This historical record may serve as a logical timeline that defines the order of concurrent operations. Files and chunks, as well as their versions, may be uniquely and eternally identified by the logical times at which they were created. Master 130 may append log records to the end of previous log records, possibly in batches. ~~Operation~~ Operational log 440 may allow the state of master 130 to be updated simply, reliably, and without risking inconsistencies in the event of a master 130 crash.

Please replace paragraph [0051] with the following rewritten paragraph:

Because of the importance of ~~operation~~ operational log 440, master 130 may store it reliably and not make changes visible to clients 110 until metadata modification are made persistent. ~~Operation~~ Operational log 440 may be replicated on multiple master replicas and respond to clients 110 after the log record for an operation is written. A log record may be considered written after it has been flushed to persistent local memory by master 130, as well as all master replicas.

Please replace paragraph [0052] with the following rewritten paragraph:

Master 130 may restore its file system state by replaying ~~operation~~ operational log 440. To minimize startup time, ~~operation~~ operational log 440 may be kept reasonably small. Master 130 may checkpoint the state whenever ~~operation~~ operational log 440 grows beyond a certain

size. Thus, when master 130 starts up, it can restore its state by beginning from the most recent checkpoint and replaying only the log records after the checkpoint file. The checkpoint may be written as a compact B-tree that can simply be mapped into memory and used to serve namespace lookup requests without further parsing. This speeds up recovery in the event of a failure and, thereby, improves the availability of the file system.

Please replace paragraph [00130] with the following rewritten paragraph:

If a write operation overwrites an existing range of the chunk, however, the first and last blocks of the range being overwritten may be read and verified. The write operation may then be performed and the new checksums may be determined and logged. If the first and last blocks are not verified before overwriting them partially, the new checksums may hide corruption that ~~exists~~ exists in the regions not being overwritten.